CS112 - Fall 2022

Lab20

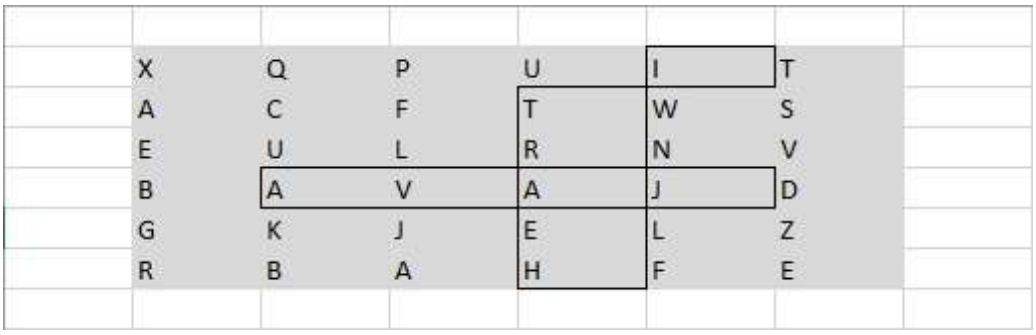
Instructor: Paul Haskell

INTRODUCTION

In this lab, you will work with multidimensional arrays and with ArrayLists.

Word Search

You know those word search puzzles, where you are given a grid of letters and you have to search for hidden words?



	X	Q	P	U	I	T
	A	C	F	T	W	S
	E	U	L	R	N	V
	B	A	V	A	J	D
	G	K	J	E	L	Z
	R	B	A	H	F	E

Your first assignment this week is to finish implementing a program **WordSearch.java.** The program reads in a square array of characters from the keyboard i.e. from `System.in`. Each line of input contains one line of the WordSearch puzzle. You can save the characters to a two-dimensional array. Then you read a "search word" from `arg[0]` and search for it in the array. To keep things simple, only search forwards, backwards, upwards, and downwards. Not diagonally.

For your program, you should convert the input puzzle from capital letters to lowercase letters. If you find a match for the search word, convert only the letters in the puzzle that contain the search word to UPPERCASE. And at the end, print out the resulting puzzle. If you do not find a match, do not print the puzzle but only print out "NO MATCH" to `System.out`.

The **CourseInfo/Lab20** directory in the CourseInfo repository has a portion of **WordSearch.java** written already. You have some additional methods to fill in. There are a few hints in the program comments about how you could approach this.

List Merge

This program, **Merge.java**, might be a little easier. Three command line arguments give the names of three input text files, each of which contains an already-sorted (increasing order) list of integers. The program should read the three lists (into ArrayLists?) and print out to `System.out` a merged sorted (increasing order) list that combines all elements of the three input lists.

Since the input lists are sorted already, it should be easy to combine them in sorted order. Hint: just look at the first not-yet-output elements of each list. But be warned: not all the lists are necessarily the same length.

Error handling is as usual: if there is a problem with the user input, print 'ERROR' to `System.err`, along with an explanation of the error. If there is an error, print nothing to `System.out`.

Reminder

Put all your files in your **Lab20** directory and push to GitHub before the deadline. This assignment must be turned in before 11:59pm Friday November 11.

Conclusion

You should have gained some experience working with more advanced data structures. Hopefully you can benefit from the convenience and capabilities of the built-in data structures, and hopefully you can design your own data structures when you need something customized. Some of this code will be useful for Project02.

Grading Rubric

WordSearch.java is worth 15 points: 3 points each for 5 test cases

Merge.java is worth 15 points: 3 points each for 5 test cases